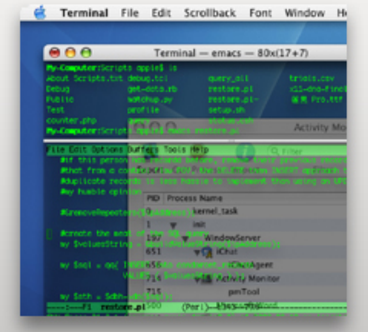
Extending Your Application to Run on an Intel-based Macintosh

Session 003

These are confidential sessions—please refrain from filming or taking pictures.



General Sessions

Extending Your Application to Run on an Intel-based Macintosh

Eric Albert and Ronnie Misra
Core Technologies

http://developer.apple.com/wwdc2005/

Moving Any Application

- Build it
  - Move to GCC 4.0
  - Make the build environment work
  - Adapt to a few small API changes
- Run it and see what breaks

How hard is this?

Moving Any Application

Not very hard at all.

Target: Firefox

4 million lines of code

Building Fink

Download Fink and start the install

```
Checking system... i386-apple-darwin8.1.0
Fink is currently not supported on x86
Darwin. Various parts of Fink hardcode
'powerpc' and assume to run on a PowerPC
based operating system. Use Fink on this
system at your own risk!
```

Building Fink

- Look for that message in the Fink code and find this in perlmod/Bootstrap.pm:

```
} elsif ($host =~ /^i386-apple-darwin(6\.[0-6]|[7-9]\.)/) {
   &print_breaking("Fink is currently not supported on x86 ".
      "Darwin. Various parts of Fink hardcode 'powerpc' ".
      "and assume to run on a PowerPC based operating ".
      "system. Use Fink on this system at your own risk!");
   $distribution = "10.2";
```

- Add a new case:

```
} elsif ($host =~ /^i386-apple-darwin8\.1\.0/) {
   &print_breaking("This brand new system is still being tested " .
          "but should work.");
   $distribution = "10.4-transitional";
```

Building Fink

Next error

```
In file included from dump.c:37:
/sw/src/fink.build/dpkg-bootstrap-1.10.21-201/dpkg-1.10.21/include/
parsedump.h:33: error: array type has incomplete element type
```

- Needs a fix for GCC 4.0

That's all!

Firefox

Step 1: Recognize the Platform

- Find code that recognizes which platform it's running on and update it to recognize Intel-based Macintoshes

Step 2: Update to New APIs

- If necesssary, adopt new APIs:
  - Opaque AliasRecords
  - QuickDraw picture bounds
  - Host-endian Core Graphics pixel formats

Step 3: Swap as Needed

- When you have to swap data, use the APIs in NSByteOrder.h, Endian.h, CFByteOrder.h, or OSByteOrder.h
- Don’t add #ifs for `__BIG_ENDIAN__` or `__LITTLE_ENDIAN__` when adding swapping code
  - The compiler will optimize out the swapping calls when they aren’t needed

Step 4: Adopt Build Configurations

- Use Xcode's build configurations
  - Modify makefiles and scripts to handle different output directories for each configuration

Step 5: Adhere to the Mac OS X ABI

- Don’t use regparm today
- Maintain 16-byte stack alignment

Step 6: Load the Right Plug-ins for the System

- An application can only load plug-ins of the same architecture as the application
- Don’t try to load CFM or Mach-O PowerPC plug-ins when running natively on Intel

Step 7: Read/Write Data in the Right Endianness

- Some data is always big-endian
  - Low-level font data
  - ColorSync profile element data
  - Finder information accessed via low-level APIs
  - Quickdraw picture bounds

Steps to Follow

- Move to GCC 4.0
- Move to Xcode or correct any problems with your makefiles or build tools
- Run and test your application

Other Common Issues

- File format incompatibilities
- Networking issues
- Integer divide by zero
- OpenGL, CoreGraphics, and QuickDraw color problems
- Undefined or implementation-defined code
- Custom resources, Apple events, or pasteboard types
- Remember to test the result on both PowerPC- and Intel-based Macintoshes

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Moving to gcc 4.0	Session 404 Wednesday 5:00PM
Code with Apple's Image and Signal Processing APIs for Maximum Performance	Session 110 Friday 10:30AM
Swimming with Shark	Hands On Friday 2:00PM
Universal Binary Lab	Lab Every Day

Q&A

Matthew Formica
Developer Tools & Cocoa Evangelist
mformica@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference